(1)

A RAND NOTE

DTIC FILE COPY

AD-A230 216

User Guide for the RAND Database Handling (DBH) System

Walter V. Hobbs, Donald E. Lewis

October 1990

DTIC
ELECTE
DEC 13 1990
S B D

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

90 12 17 095

RAND

The research reported here was sponsored by the United States Air Force under Contract F49620-86-C-0008. Further information may be obtained from the Long Range Planning and Doctrine Division, Directorate of Plans, Hq USAF.

The RAND Publication Series: The Report is the principal publication documenting and transmitting RAND's major research findings and final research results. The RAND Note reports other outputs of sponsored research for general distribution. Publications of The RAND Corporation do not necessarily reflect the opinions or policies of the sponsors of RAND research.

Published by The RAND Corporation
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138

A RAND NOTE

N-3104-AF

User Guide for the RAND Database Handling (DBH) System

Walter V. Hobbs, Donald E. Lewis

October 1990

Prepared for the
United States Air Force

RAND

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

PREFACE

This Note is a user guide for the RAND Database Handling (DBH) System developed to operate on the PACAF Cryptologic Support Group SAAFE database used to support a project entitled "Air Base Vulnerabilities of Potential Adversaries in the Pacific Basin," performed under the Theater Force Employment program of Project AIR FORCE, an OSD-supported federally funded research and development center. This study was initiated by the Office of the Deputy Chief of Staff, Intelligence, Headquarters PACAF. The study is an assessment of the vulnerabilities of Far East Military District air bases and their infrastructure to offensive counter air missions carried out by the Pacific Air Forces.

This user guide should assist the DBH user in operating the various parts of the system. It assumes the reader has a basic knowledge of UNIX and its utilities; therefore, they are not defined in the text.

This Note should be of interest to operations analysts, intelligence specialists, data handlers, and others involved in developing and assessing databases.

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

SUMMARY

The RAND Database Handling (DBH) System was developed to support the SAAFE database maintained by the PACAF Cryptologic Support Group. This consists of five separate programs, each of which the user executes individually. Each program accomplishes a specified task relative to the overall requirement of processing data and generating reports.

First, raw data are sorted and duplicate entries and unnecessary lines are removed by the "sort" program. Then the resulting file is broken into several manageable pieces, each a UNIX file, by the "split" program. An identifying field is also attached to each data entry by the "split" program.

Next the data file is examined by the "fix" program and corrections are made to the data for misspelled words and format errors, etc. Although most of the errors and inconsistencies can be corrected by the program, some conditions must be manually changed by editing the individual data files.

Once through the "fix" program, the data are ready for the analyst to use. The next program, called "select", generates a smaller file of particular interest.

The final program is named "sdcalc". It is used to compute specified values and generate analytic reports.

CONTENTS

PREFACE iii

SUMMARY v

FIGURE AND TABLE ix

Section

I. INTRODUCTION 1

II. DESIGN CONSIDERATIONS 2
Requirements and Limitations 2
Software Guidelines 3

III. FILE FORMAT 4

IV. SYSTEM OVERVIEW 6

V. SORT PROGRAM (SDSORT) 8

VI. SPLIT PROGRAM (SDSPLIT) 9

VII. FIX PROGRAM (SDFIX) 10

VIII. SELECT PROGRAM (SDSEL) 13

IX. CALCULATE PROGRAM (SDCALC) 16

X. FINAL REMARKS 18

Appendix: EXAMPLES OF OUTPUT LISTINGS
FROM SDCALC 19

FIGURE

1. Database handling (DBH) system overview 7

TABLE

1. Field content and format 5

I. INTRODUCTION

The RAND Database Handling (DBH) System was developed to operate on the PACAF Cryptologic Support Group SAAFE database for use by Headquarters Pacific Air Forces. When the project started, it was known that the customer might request the software for delivery and subsequent use. However, a short time schedule precluded optimization of the software for the customer's end use. We believe, however, that in its current state, the software produced for the RAND users would also be understandable to users outside of RAND.

The user interface was made as friendly as possible, given the constraints of the UNIX operating system. That is, some users consider UNIX to be not too friendly, at least compared with, for example, the Macintosh operating system. To make DBH more user-friendly, all the DBH programs offer a "-help" option that shows online help information about each DBH program.

Section II discusses some points of interest relating to the design of the DBH software. Section III presents the format of the data file on which the DBH software operates.

The remainder of the document discusses the use of the DBH software. Section IV gives a brief overview of the system. Sections V through IX are the user level details relating to each major part of the DBH software. Section X presents some final remarks regarding the Database Handling System. Finally, the appendix contains examples of different output listings produced by the software.

II. DESIGN CONSIDERATIONS

REQUIREMENTS AND LIMITATIONS

The following requirements and limitations dictated to a large degree the design of the software for the project:

- A quick start with somewhat unknown data. A software capability was needed as soon as possible. Although that is not unusual in software development circles, in this case we received the data concurrently with what should have been the start of software development. This was our first look at the data; as was suspected, although there was a document that described the data, the actual data were not well defined for computer use.

  It was later found that in fact the data were originally designed for manual manipulation, not for computer analysis. Therefore, much of what we received was readable and understandable by humans but required a considerable amount of modification to be usable by the software. Furthermore, the data contained 80 percent or more of free form, variable length written text.
- Flexibility. Since the overall understanding of the data could not be known ahead of time, details and formats of the analytical reports were also unknown. Therefore, it was necessary to be able to modify the software as the analysis proceeded and the analytic requirements emerged.
- Fairly good computer performance on a small machine. The computer equipment available for use was a SUN 2/120, which was later replaced with a SUN 3/50. Although both of these computers were considered to be adequate for the job, they are fairly "slow" among computers that might be candidates for this type of work.
- The computer operating system would be the UNIX Operating System, in wide use at RAND.
- The nature of the data necessitated that the work be done in a secure environment.

- There was a good chance that the software would have to be delivered to the customer.

SOFTWARE GUIDELINES

As a result of the above design considerations, we formulated the following software design guidelines:

- The software would be portable to other UNIX machines.
- We would avoid the use of proprietary application software—e.g., commercial Database Management System (DBMS) software.
- The software would use the standard UNIX utilities and public domain programs.
- The software would have as "friendly" a user interface as possible given the development time constraints.
- The software would be composed of functionally separate programs that manipulate the data, with some manual editing necessary.

Minimum system requirements

UNIX Operating System (SunOS, any version, including utilities)

PERL—available to public

Recommended additions

RAND 'E' editor—public domain

LESS—public domain

III. FILE FORMAT

The final file format is basically that provided by the customer. Where ambiguities would affect computer operations, we extended the definitions to be more restrictive. In all cases we believe these extended definitions do not affect or restrict the use of the data in any way.

The data file (sometimes referenced as the database file) physically consists of several UNIX files. These are standard ASCII data files, and there is no machine structure imposed on them. The names of the files follow a convention so that the user can easily manipulate them, either when executing the programs or when it is necessary to manually edit a file with a text editor. This also helps when using other UNIX utilities, such as 'grep' or 'less'.

The data file is logically composed of several records, or entries, where each entry can vary in size. An entry is composed of several lines, and each line can vary in length.

There are two types of lines in an entry: Header lines and Free Field lines. An entry includes at least one Header line; Free Field lines are optional. Typically an entry will include several Header lines and many Free Field lines.

A Header line is composed of "fields" (a contiguous string of characters with no embedded blanks). There are one or more space characters (blanks or tabs) between fields—e.g., these space characters are the field separators. The order of the fields is significant, but (with one exception) the spacing between fields is arbitrary. A Header line contains the following fields, in order:

Date Time Unit Base Nr Type Msn SecMsn Weekday Id

The content and format of the fields in a Header line is shown in Table 1.

A typical Header line might look like:

880612* 0900–1400 U123 BABC 7 F-16A TRN (AAA,BBB) FRI X1234

The first Header line in an entry is known as the "Main" Header line, and defines the start of an entry. Subsequent Header lines in the entry are known as "Secondary" Header lines. They do not have the asterisk at the end of the Date field. Secondary Header lines immediately follow the Main Header line in an entry. Following any

Table 1

FIELD CONTENT AND FORMAT

Name	Format	Content
Date	YYMMDD*	Numeric Year (YY), Month (MM), Day (DD). The asterisk indicates this is a Main Header line; a Secondary Header line will have a blank space instead.
Time	NNNN-NNNN	Numeric military times; Up-time (NNNN), a dash (-), and Down-time (NNNN). The times may be "...." instead of numeric characters. An optional "." may precede the first numeric character.
Unit	XXXX	Alphanumeric field of varying lengths.
Base	XXXX	Alphanumeric field of four characters.
Nr	NN	Numeric field of varying lengths.
Type	XXXX	Alphanumeric field of varying lengths.
Msn	XXXX	Alphanumeric field of varying lengths.
SecMsn	XXXX	Alphanumeric field of varying lengths. Has one or more three or more character words separated by commas inside parens, with NO embedded blanks, e.g., "(AAA,BBB,CCC)".
WeekDay	(AAA)	Weekday, alpha field of three characters, in parens, e.g., "(MON)" or "(WED)".
ID	XXXX	Alphanumeric field of varying lengths (optional).

Secondary Header lines are the Free Field lines. *Note that the Date field must start in the first character position in the line. The first two fields (Date and Time) must be separated by exactly one blank space for the Main Header line only.*

A Free Field line starts with one or more space characters and has a single field (a line type identifier) followed by arbitrary text of any type. There are several different types of Free Field lines—e.g., RMK, ALT, CMNT, MSN, and NOTE. A typical Free Field line of type "RMK" might look like:

RMK Text here that could be of varying lengths on one line.

The entries in the data file are sorted by the Date and Up-Time fields of the Main Header line.

There is a blank line between entries, which is for readability only. It does not affect the programs since a Main Header line defines the start of an entry.

IV. SYSTEM OVERVIEW

The DBH system consists of five separate programs written in PERL (see Fig. 1), each of which the user executes individually. Each program performs a specified task relating to processing the data and generating reports for the analysts.

First, the raw data file is sorted and duplicate entries and unnecessary lines are removed by the sort program "sdsort".

The resulting data file is split, using "sdsplit", into manageable sized pieces, each a UNIX file, that together constitute the overall data file. At the same time, an identifying field is automatically added by "sdsplit" to each entry, at the end of the Main Header line.

Next the data file is examined by the fix program "sdfix", and corrections are made where possible, with a list of exceptions output to a file (the exception listing). While most of the errors and inconsistencies can be corrected by the "sdfix" program, some conditions must be manually changed by editing the individual data files (i.e., the UNIX files).

Now the data file is ready for use by the analysts. The select program "sdsel" is used to generate a smaller file of particular interest.

The calculate program "sdcalc" is used to compute specified values and generate the analytical reports. This program, which is the only one that does not produce a database file, can execute using either a subset data file produced by the select program or the entire data file, the difference being a longer execution time in the latter case.

Not all of the abovementioned programs need be used, or in the above order. For example, the "fix" program may be used repeatedly (followed by editing of the data files) until the data are in suitable form. Also the select program "sdsel" need not be used if the calculate program "sdcalc" is to cover data that span an entire data file. As a matter of fact, if the "Raw Data File" is in order and consistent, it can be used as input to "sdcalc". Both "sdsel" and "sdcalc" have the same format; only the size differs in some cases.

To provide assistance to the user, each of the individual programs will accept an option of the form "-help", which will provide on-line information about that program. In addition, help is available for the UNIX operating system commands by means of the "man" command.

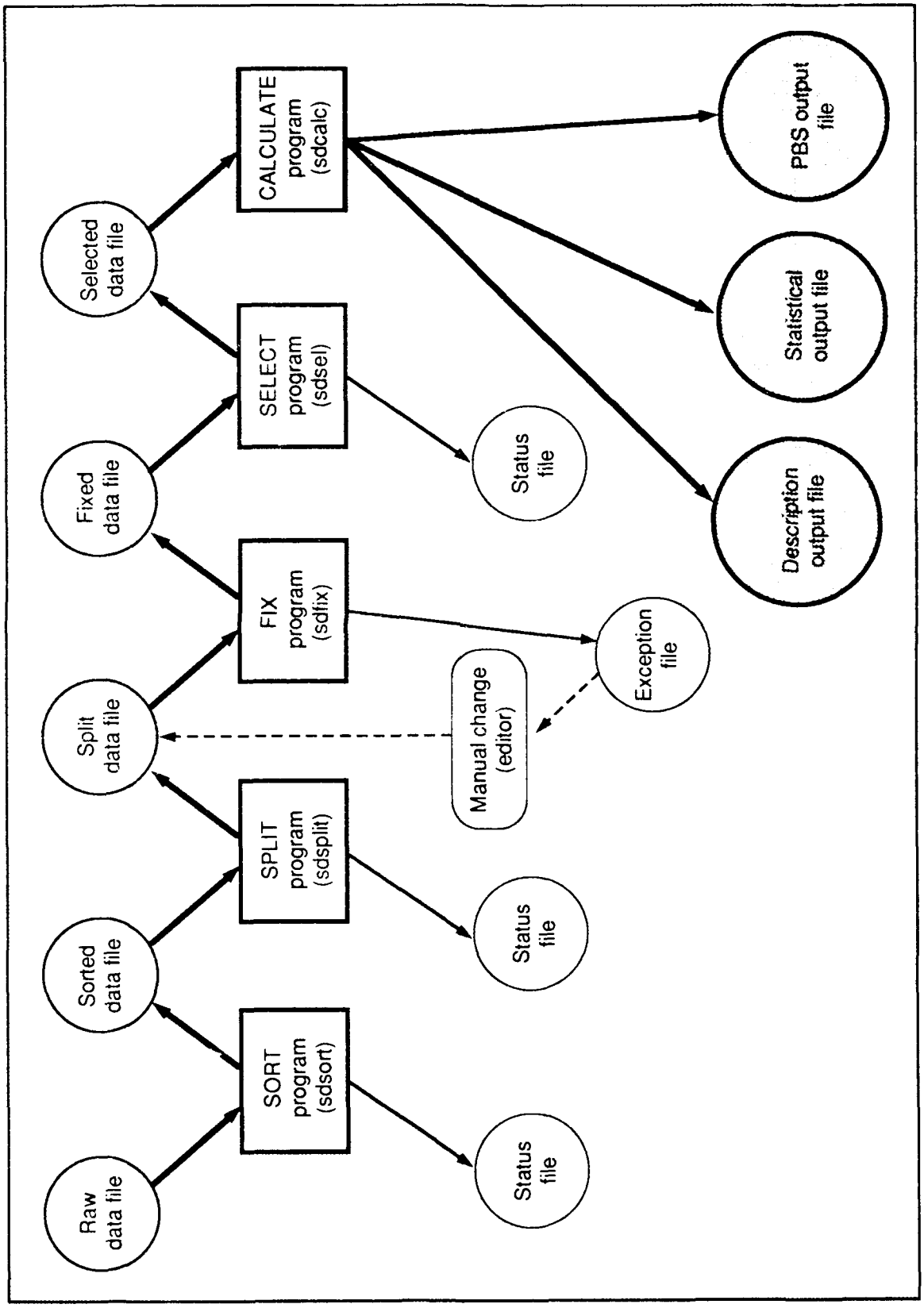

Fig. 1—Database handling (DBH) system overview

V. SORT PROGRAM (SDSORT)

- Purpose: To sort a data file and delete duplicate entries.
- Usage: sdsort options database_file

Options	Meaning
-help	Print help information.
-n	Does not sort; just do the first pass and print summary.
-x	Prints a status list to a file; default is to STDOUT, e.g., to screen unless redirected.

With no options it sorts "database_file" to "database_file.S".

- Input: A database file. This is typically the "raw" data, which might be out of order and/or contain duplicate entries.
- Output:[1] A sorted database file with no duplicate entries.
- Typical command line: To sort the raw data file "db.raw", putting the sorted data on file "db.raw.S" and the status listing on file "db.raw.X", the command would be:

sdsort -x db.raw

- General: Uses standard UNIX utilities (SORT, UNIQ) to sort and remove duplicate entries.

  The sorted order is on date and up-time fields of the Main Header line.

  The definition of a duplicate entry is as follows. Two entries are identical if the Main Headers of the entries are identical and both entries have the same number of other lines.

[1] A Status File produced by the "-x" flag is also generated. It contains the following information for both the source and output files: (1) number of entries, (2) maximum entry size in lines, (3) total lines in file.

VI. SPLIT PROGRAM (SDSPLIT)

- Purpose: To split a large database file into a convenient number of UNIX files and add the ID fields to the first line (Main Header) of all entries.
- Usage: sdsplit options database_file

Options	Meaning
-base name	"Name" is the basename of new files; default is "x".
-id opt	ID option: "add"—add ID if entry has none, "new"—replace all entries with new values (default).
-help	Print help info.
-x	Prints a status list to file; default is to STDOUT.
-size NNN	Minimum size in bytes of new files. May have "k" suffixed to mean "kilobytes". The default is 500K bytes.

Option names may be shortened to a nonambiguous abbreviation, e.g., "-b" for "-base".

- Input: A database file. This is typically a single large file.
- Output:[1] A set of smaller database files that, taken together, constitute the logical database. ID fields are added.
- Typical command line: To split the data file "db.big" into several UNIX data files named "newNNNN" (where "NNNN" is the number of the year and month of the first entry in each UNIX data file), putting the status listing on file "new.X", and generating new ID fields in each entry, the command would be:

sdsplit -base new -x db.big

- General: The program "sdsplit" splits a database file into one or more new files, starting with the first entry of a month, with the file size being at least that specified in the "-size NNN" option, except for the last file.

[1]If a Status File is generated, it contains the following information for each of the new generated files, the input file, and a sum for all the generated files: (1) number of entries, (2) total number of lines.

VII. FIX PROGRAM (SDFIX)

- Purpose: To check entries in database and modify for consistency as required.
- Usage: sdfix options database_file

Options	Meaning
-fix	Generate new database file. Entries will include only valid main header lines and other valid lines. Nonvalid lines and messages will be on the exception list.
-help	Print help info.
-HDR	List all header lines.
-pass	Like "-fix" except ALL nondeleted entries from the input database file are passed to the new database file including any invalid lines. Use caution. The exception list is still produced.
-xcep	Print exception list to file; default is to STDOUT.

Option names may be shortened to a nonambiguous abbreviation, e.g., "-x" for "-xcep".

- Input: A database file. This is typically one of the smaller subset files that are a part of the logical database.
- Output: A database file and an exception list file.
- Typical command line: To correct the data file "dbunk", putting the new data on file "dbunk.N" and the exception listing on file "dbunk.X", the command would be:

sdfix -f -x dbunk

Note that if the input file ends in the form ".AAA", where AAA is any number of alpha characters, the output file name will be the same but stripped of the ".AAA" suffix.

- General: With no options it checks the database file and shows bad lines and corrections on exception list (STDOUT).

The program does a large amount of work on the database, but a considerable amount of manual editing is, for example, also necessary (because of the nature of the data) to convert the structure of the data to a form that can be used by the analytical software.

The software tries to fix those inconsistencies in the database that have been seen to occur on several entries and that can be fixed by machine with a high degree of confidence in correct results. Items that are corrected are as follows.

Main Header line:

Weather entries: If there are four fields or less, will completely reconstruct the entry, adding dummy data fields.

SecMsn: Supplies missing commas and parens. Deletes embedded blanks or duplicate commas. Puts space between SecMsn and Weekday, if none exists.

Verifies the correct order of the entries—e.g., that they are on the date and up-time fields of the Main Header lines.

Verifies the format of the date and time fields and corrects if possible.

Time field corrections are made when there are trailing periods or alpha characters, when the down-time is missing, and when the groups of periods for up-time are less than four.

One space is put between the Date/Time fields if there is more than one or none.

Base must be four alphanumeric characters or "TGT".

Number of fields must be between seven and nine (not counting the ID field added by the Split Program). If there are eight fields, the last field must have parens on it (i.e., either SecMsn or Weekday fields exist). If there are seven fields, there must NOT be any SecMsn or Weekday fields.

On lines with two to four fields (except weather entries), the program generates missing fields using default data—e.g., "....".

Secondary Header line:

If a Secondary Header line follows a blank line, this is flagged by printing a warning message, as this condition may be an error. That is, this line may be a Main Header line that has a format error.

The same checks are made as in the Main Header line except that the date/time order is not enforced and the detailed format checks are not made.

Free Field lines:

The Free Field tag is checked against a list of acceptable tags and the spelling is verified or corrected, if possible.

Several types of lines are reformatted. The tags are forced to upper case, and a standard amount of leading spaces is put at the front of the line. These include: SOU, C/S, RMK, CMNT, ALT. Several ways of spelling "RMK" are anticipated.

Extraneous characters (^Z, Linefeeds, etc.) are removed, and "bad" lines, i.e., those with only dashes or "qed" lines, are deleted. Trailing blanks are deleted. A single blank line is forced between entries.

VIII. SELECT PROGRAM (SDSEL)

- Purpose: To select entries from a database file based on a user-specified set of parameters.
- Usage: sdsel Select_Params options database_file ... Select_Params are key-value pairs that designate the criteria for selection of entries. Keys are:

date, time, ud, base, nr (or no, num), ac, msn,
smsn, wkday, free-field (or ff).

Key names may be shortened to a nonambiguous abbreviation—e.g., "b" for "base".

Pattern matching can be used for "value" if desired; quotes may be necessary to protect some characters from the UNIX shell.

Options	Meaning
-help	Print this help info.
-HDR	Prints the matching Header lines of selected entries.
-HALL	Print all Header lines of selected entries.
-o outfile	Generates new database file called "outfile".
-union	Union of Header and Free Field selection parameters. Consider the group of Free Field selection parameters combined with the group of Header selection parameters so that if any parameter is satisfied the entry is selected.
-v	Inverse option, selects all entries that do NOT match the criteria.

Option names may be shortened to a nonambiguous abbreviation—e.g., "-u" for "-union".

If no options are specified, a "-d" is required before the database file(s).

- Input: A database file or files. This is typically the set of files that constitute the logical database.

- Output: A database file consisting of only those entries selected. A status file[1] that ends in a .X is also generated.
- Typical command line: To select entries about F-16 aircraft, with a mission of "tac" using data file "db.8701", and putting the output listing on file "out_9", the command would be:

sdsel ac F16 msn tac -o out_9 bd.8701

- General: The program works by passing the database and extracting those entries that match the selection criteria. The selection can be made on any field in a Header line (except the ID) or any value in a Free Field line. If there is more than one Header line in the entry, the entry will be selected if there is a match on the data in any Header line.

  The selection parameters are one or more key-value pairs where the keys refer to a Header line field or a Free Field line. The value represents what is acceptable for these keys in order that the entry be extracted. For example, a key-value pair of:

base HOME

  refers to all entries having a "base" field that equals "HOME", or "Home", or "home", i.e., case is not significant. Multiple choices of values for a key are connected by commas, with no embedded blank spaces, as:

base HOME,AWAY,GONE

  which refers to all entries having a "base" field that equals "HOME", "AWAY", or "GONE", in any combination of cases. As an example, the command to select all entries that have any of the above values for "base" and have an msn field value of "TAC" from the database file "testdb" would be:

sdsel base HOME,AWAY,GONE msn TAC -d testdb

---

[1]This file provides the following information: (1) an image of the command line, (2) number of selected entries, (3) total number of selected lines.

The user can specify a Free Field selection by using the key "free-field", or just "ff", followed by the value, as:

ff some-string

Note that no embedded blank spaces are in "some-string". If it is desired to search for a string that does include spaces, which can be done only on Free Field lines, the value must be enclosed in quotes as:

ff "some string or other"

The user can specify values as patterns.

The user can request the inverse (all entries that don't match) with the "-v" option. That is, all entries that do not fit a given selection criterion will be related. This is useful when there are uncertainties about the database.

IX. CALCULATE PROGRAM (SDCALC)[1]

- Purpose: To calculate and report different types of occurrences in selected entries of a database file.
- Usage: sdcalc Select_Params options database_file ... Select_Params are key-value pairs that designate the criteria for selection of entries. Keys are:

date, time, ud, base, nr (or no, num), ac, msn,
smsn, wkday, free-field (or ff).

Key names may be shortened to a nonambiguous abbreviation—e.g., "b" for "base".

Pattern matching can be used for "value" if desired; quotes may be necessary to protect some characters from the UNIX shell.

Options	Meaning
-help	Print this help info.
-o outfile	Names output files called "outfile" (Description Output File), "outfile.X" (Statistical Output File), which are two of the files generated by this program.
-pbs[2]	Generates the PBS Output File.

If no options are specified, a "-d" is required before the database file(s).

If no output file is specified, the output is produced on the screen.

Option names may be shortened to a nonambiguous abbreviation—e.g., "-h" for "-help".

- Input: A database file. This is typically the output from an "sdsel" run.

[1]See the appendix for examples of the output listings.
[2]If the -pbs option is selected, the only output generated is the PBS Output File.

- Output: A Descriptive Output File and a Statistical Output File or a PBS Output File.
- Typical command line: To generate the normal calculations on data about F-16 aircraft, from base "abcd", using data file "db.8701", putting the output listing on file "out_1", the command would be:

sdcalc ac F16 base abcd -o out_1 db.8701

- General: This program uses the same selection process as the Select program. The selection process is necessary in this program because entries produced by "sdsel" may contain entries with Header lines that are not of interest for a given run of "sdcalc".

  For the -o option, the two types of files produced are descriptive output and statistical output. The descriptive output file is a single format listing, and the statistical output file is generated in a variety of formats.

  The descriptive output file includes some information from the Header lines plus deployment base information from the Free Field lines, when available.

  The statistical output file includes the statistical information report, and other reports on sorties listed by day of week, hour of day, elapsed time, month, sorties per day groups, mission, and sub-mission.

  The user can request the PBS listing by specifying the month and giving the PBS option "-pbs" as follows. A typical command, requesting a PBS listing for June 1987, with output on file "out.file", using data file "db8701", would be:

sdcalc date 8706 -pbs -o out.file db8701

  Internal calculations use real numbers, but all output numbers on the PBS listing are rounded. This could result in a slight apparent error in the listing.

X. FINAL REMARKS

The DBH program was developed to operate on the PACAF Cryptologic Support Group SAAFE database utilizing currently available tools—e.g., UNIX and PERL. This allowed the rapid development of this software with the flexibility necessary for the task at hand. The major requirements of the task were to:

1. Correct inconsistencies in the original database.
2. Verify and sort data in proper order.
3. Select specific datasets based on user input.
4. Calculate usable statistical data.

Having the above tools allowed the DBH software to be completed within the required time.

Appendix

EXAMPLES OF OUTPUT LISTINGS FROM SDCALC

A. Descriptive output is produced on the "outfile". A line of data is derived from each Header used, as follows:

```
870601  UNIT   BASE  DBAS  3   F16A  TES   (AAA,SSS)     X71
870602  UNIT   BASE  DBAS  2   F16   SAM   (BBB,SSS)     X72
```

The data items above are, in order, date, unit, base, D-base, (dispersal base), number of aircraft, type of aircraft, mission, sub-missions, and the ID. All data are direct from the Header line and Free Field lines with no modifications.

B. Statistical output is produced on the "outfile.X". The data from all entries are presented in different ways, as follows.

Summary Information:

Nr Sorties: 20 (Est: 3) Nr Days: 9 Sorties/Day: 2.22

Nr Entries: 10 Nr Hdrs: 12

If an entry has a Header with the "NR" field that does not contain digits, the number of aircraft for that Header is estimated at one. In the above example, the datum "Est: 3" means that three entries had the number of aircraft estimated for this reason.

The datum "Nr Days: 9" means that there were entries that referenced nine different days in the run.

Sorties by day of week. The number and percent of total on each day of the week.

Sorties by day of week

DAY	NR	%_TOTAL
SUN		
MON	3	15.0
TUE	2	10.0
WED	15	75.0
THU		
FRI		
SAT		
TOTAL:	20	

Sorties by hour of day. The number of sorties by Greenwich mean time and local time that took off, landed, or were in the air, as defined by "uptime", "downtime", and "any".

Sorties by hour of day

ZULU	LOCAL	UP	DOWN	ANY
16	1			
17	2	1		1
18	3		1	1
...	..	..	..	..
13	22	3		3
14	23			3
15	24		3	3
TOTALS:		15	15	41

Sorties by Elapsed Time in Hours. The number of sorties for different elapsed times, with percent of total and cumulative percent of total.

Sorties by elapsed time in hours

ET	NR	%_TOTAL	CUM_%
1	3	27.3	27.3
2	7	63.6	90.9
3			
4			

O ...
21
22
23
TOTAL: 11

Sorties by month. The number of sorties listed by month, with the percent of total. The range of months here is January 1987 through April 1989.

Sorties by month

MONTH	NR	%_TOTAL
8701		
8702	5	25.0
8703		
8704		
8705		
8706	5	25.0
8707	10	50.0
8708		
..		
8904		
TOTAL:	20	

Sorties per day, frequency. The number of sorties per day and the frequency of those occurrences, with percent of total and cumulative percent of total.

Sorties per day, frequency

NR	FREQ	%_TOTAL	CUM_%
1	4	44.4	44.4
2	2	22.2	66.7
3	2	22.2	88.9
4			
5			
6	1	11.1	100.0
TOTAL:	9		

Sub-Msn, %_Total Sorties. The percent of total sorties that refer to each sub-mission, sorted by largest percent.

Sub-Msn,	%_Total Sorties
SSS	100.0
CCC	45.0
AAA	15.0
GGG	10.0
...	...

MSN %_TOTAL Sub-Msn %_MSN_Total. The percent of total sorties that refer to each mission, sorted by largest percent. Within those groups, the percent of sorties that refer to each sub-mission, sorted by largest percent.

MSN	%_TOTAL	Sub-Msn	%_MSN_Total
TES	40.0		
		SSS	100.0
		AAA	37.5
		CCC	37.5
		BBB	25.0
KIK	35.0		
		SSS	100.0
		CCC	71.4
		FFF	14.3
...	...	...	

C. PBS output (shortened for this document) is shown below. (The normal listing is wider to accommodate 31 days of the month.) It lists for each PBS the number of sorties flown in missions PRO, TAC, and OTH (all other missions). It also gives totals by mission and overall total by PBS, as well as overall totals at the end of the file.

Sorties (PRO,TAC,OTH) by PBS for Day of Month: 8706

PBS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	31	SUB	TOT
101	2	.	.	.	.	.	.	.	.	7	.	.	.	.	5	.	...	.	14	
	.	.	.	3	.	.	.	.	.	.	.	.	.	.	.	.	...	.	3	
	.	.	.	.	.	.	.	8	.	.	.	.	.	.	.	.	...	.	8	25
102	6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.	6	
	.	.	.	.	.	.	.	.	.	.	.	8	.	.	.	.	...	.	8	
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.	.	14
...																				
909	6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.	6	
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.	.	
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.	.	6
PRO	.	.	.	.	40	.	.	.	.	7	.	.	.	.	5	.	...	.	52	
TAC	.	20	.	.	.	.	.	.	.	.	.	8	.	.	.	.	...	.	28	
OTH	60	.	.	.	.	.	.	8	.	.	.	.	.	.	.	.	...	.	68	148
TOT	60	20	.	.	40	.	.	8	.	7	.	8	.	.	5	.	...	.	148	